

AD-A135 452

ANNUAL SCIENTIFIC REPORT FOR GRANT AFOSR-81-0205(U)
TEXAS UNIV AT AUSTIN DEPT OF COMPUTER SCIENCES
K M CHANDY ET AL. SEP 83 AFOSR-TR-83-0993 AFOSR-81-0205

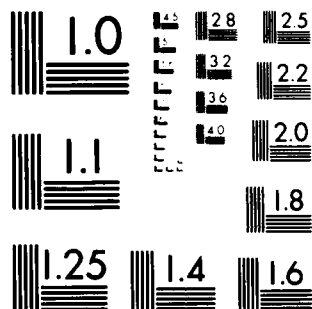
1/1

UNCLASSIFIED

F/G 5/2

NL

END
DATE
FILMED
1 84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(3)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 83 - 0993	2. GOVT ACCESSION NO. A135452	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ANNUAL SCIENTIFIC REPORT FOR GRANT AFOSR-81-0205		5. TYPE OF REPORT & PERIOD COVERED Annual, 15 JUN 82-14 JUN 83
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) K.M. Chandy and J. Misra		8. CONTRACT OR GRANT NUMBER(s) AFOSR-81-0205
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Sciences Department University of Texas Austin TX 78712		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE61102F; 2304/A2
11. CONTROLLING OFFICE NAME AND ADDRESS Mathematical & Information Sciences Directorate Air Force Office of Scientific Research /NM Bolling AFB DC 20332		12. REPORT DATE SEP 83
		13. NUMBER OF PAGES 15
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) In the last several years, the investigators have developed a number of important, practical distributed algorithms for problems such as distributed simulation, termination and deadlock detection, computing networkwide functions in a distributed manner, etc. Interest in distributed systems has spurred publications of many distributed algorithms by other researchers. The major thrust in the past year has been directed towards developing unifying frameworks, i.e., paradigms - which consolidate the known results. They have developed a theory for detecting all system properties that are (CONTINUED)		

DTIC

SELECTED

J

DTIC FILE COPY

DD FORM 1473

1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

~~U~~ ~~CONFIDENTIAL~~

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ITEM #20, CONTINUED: Stable, i.e., properties that continue to hold once they begin to hold. Examples of stable properties are termination, deadlock, etc. The work subsumes a large body of literature on termination and deadlock detection. They have similarly developed a theory of conflict resolution and an algorithm based on this theory; this algorithm subsumes all known non-probabilistic algorithms for mutual exclusion, resource sharing, dining philosophers' problem, etc. In addition, they have continued work on models of distributed systems, their verification and performance analysis.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

September 1983

Annual Report for AFOSR-81-0205

1. OBJECTIVES

The goal of this project is to help build the conceptual foundations underlying distributed information systems. A new area in computer sciences is characterized by a wealth of papers, conferences and buzzwords, because it takes time to develop the few concepts which form the basis for many apparently distinct ideas: this project is concerned with identifying these concepts and then demonstrating that many practical problems can be solved as special cases.

Our work in distributed systems proceeds in 3 steps:

Step 1: What is truly fundamental?

Identify the problem (or concept, or paradigm) that subsumes a number of apparently distinct ideas. This is, in many ways, the most creative part of our work.

Step 2: Solve the fundamental problem.

Develop efficient algorithms to prove the correctness of the fundamental problem and prove its correctness.

Step 3: Apply the solution to the fundamental problem to many practical problems.

In this step we demonstrate that solutions to many important problems can be derived as special cases of the fundamental unifying concept.

We next give some examples of the work carried out in the last year.

Approved for public release;
distribution unlimited.

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFOSR)
NOTICE OF TRANSMITTAL September 1983
This technical report has been reviewed and is
approved for public release IAW AFR 100-12.
Distribution is unlimited.
MATTHEW J. KERPER
Chief, Technical Information Division

2. WORK IN '83 - '84

2.1 Stable Properties in Distributed Systems

There have been over 20 papers published on detecting deadlock in distributed systems [20,26,27], detecting termination [28], counting the number of "tokens" in a ring [6] etc. Several of the published algorithms have been shown to be incorrect. Here was an area with many papers, tremendous diversity, a great deal of complexity but no common concepts.

Step 1: What is truly fundamental?

All the properties that are detected in the papers cited above are stable in the sense that once a system possesses a stable property it continues to possess that property thereafter. Once a computation has terminated, we may assert thereafter that it has terminated. Once a system has deadlocked, we may assert thereafter that it has deadlocked. The single unifying concept underlying all detection algorithms is the stable property.

Step 2: Algorithm to solve the fundamental problem.

A stable property is a property of the state of the system. Thus, the problem is to detect the state of a distributed system. The difficulty is that in the absence of a global clock, all the processes in a distributed system cannot take a snapshot of their states and channels at precisely the same instant.

Chandy and Lamport developed an algorithm which allows a distributed system to detect its global state [7].

Step 3: Applications: Special cases of the fundamental concept.

We are now showing that all the published work on detection of properties of distributed systems can be derived as special cases of our state-detection algorithm. Special cases include detection of deadlock in data bases, deadlock in communication systems and termination in distributed systems.

2.2 Sharing Resources in Distributed Systems

There have been a very large number of papers written on mutual exclusion, multiple copy consistency, resource sharing in distributed data bases and the dining philosophers problem. Here too there seems to be no single common thread unifying, what appears to be, many disparate ideas.

Step 1: What is truly fundamental?

The problem is this: resources which cannot be shared simultaneously by 2 or more processes must be shared over time - i.e. one process uses the resource for a while and relinquishes it and then another process uses it, and so on.

We think the fundamental paradigm is as follows: A network of processes share resources which are represented by colored tokens. A network may contain an arbitrary, but fixed number of tokens of a given color. From time to time a process requests a set of tokens - for instance: 2 red tokens, 3 black tokens ... After a process gets the requested tokens it holds it for some finite time and then releases it. After a process requests a set of tokens it may not request

another set until it has released the previous set. The problem is to find an efficient solution which ensures that every process desiring a set of tokens receives it in finite time. This problem subsumes the dining philosophers' problem, the mutual exclusion problem and most distributed resource management problems.

Step 2: Algorithms to solve the fundamental problem.

We have devised a scheme based on partial orderings of processes to solve the problem. The basic idea is this - for some time we give one process priority over another and then by reversing priorities over the long term we achieve symmetry over the long term though every state is asymmetric [8].

It has been shown [23] that it is impossible for an ensemble of perfectly symmetric processes in a symmetric global state to resolve conflict. Therefore the study of conflict resolution is one of asymmetry introduction. Traditionally asymmetry is introduced (1) by a central process, which resolves all conflicts or (2) by resorting to probabilistic decision making by individual processes or (3) by assigning a static global priority to each process. All traditional non-probabilistic solutions distinguish processes, either by having specialized processes or by ordering process id's.

We have devised an efficient, fair, symmetric solution to the dining philosopher and mutual exclusion problems by initially introducing asymmetry through judicious placement

of resources and maintaining asymmetry during computation [8]. The notion of asymmetry introduction is so fundamental that without its recognition the general problem of resource sharing cannot be solved.

Step 3: Applications - special cases of the fundamental concept

We have applied our solution to general resource management, mutual exclusion and the dining and drinking philosophers' [8] problem.

2.3 Verification of Distributed Systems

Here too a great deal of work has been done; see [25] for a summary.

Step 1. What is truly fundamental?

A distributed system consists of processes "hooked together" via channels. The critical question in the verification area is: Are the processes hooked together correctly? In other words, how can one prove that the network of processes will do what it is specified to do, given the specifications of the component processes? We think that the most important problem in the verification of distributed systems is modularity: how can one build large modules from small modules, or equivalently, how can one decompose a complex model into simpler ones?

Our view is different from earlier methods where a distributed system is thought of as a program. We think of a distributed system in a more abstract manner as a static network interconnecting processes specified in some mathe-

mathematical notation - our focus is on levels above program statements.

Step 2: Solving the fundamental problem.

We specify each process in terms of a predicate describing what the process expects of its environment and another predicate describing what the process guarantees to its environment. We have a simple way of determining whether what a process assumes about its environment is guaranteed by the collection of all processes [4,5].

Step 3: Applications.

Most of the last year has been spent on step 3: showing how our modular proof technique can be used to solve practical problems. One of the doctoral students supported on this program, Marty Ossefort, has applied this technique to VLSI systems, communication protocols and local area networks [9,10,24].

A survey of proof methods by two British computer scientists found our method to be elegant and useful.

2.4 Performance Analysis

A student (Elizabeth Williams) completed her Ph.D in this area. Williams built a distributed test-bed simulator and studied a number of scheduling algorithms [29].

September 1983

Uses of Travel Funds

Dr. Chandy was a member of the program committee on "Principles of Distributed Computing" in 1983; PODC is the premier conference in this area. Dr. Misra is the program chairman for PODC in 1984. Travels to the conferences and attendance at program committee meetings were paid from grant funds.

References

- [1] Chandy, Mani and Misra, J., "Distributed Computation on Graphs: Shortest Path Algorithms," Communications of the ACM, Vol. 25, No. 11, November 1982, pp. 833-837.
- [2] Chandy, Mani, Misra, J. and Haas, L., "Distributed Deadlock Detection," ACM Transactions on Computer Systems, Vol. 1, No. 2, May 1983, pp. 144-156.
- [3] Misra, J. and Chandy, Mani, "A Distributed Graph Algorithm: Knot Detection," ACM Transactions on Programming Languages and Systems, Vol. 4, No. 4, October 1982, pp. 678-686.
- [4] Chandy, Mani and Misra, J., "Proofs of Networks of Processes," IEEE Transactions on Software Engineering, Vol. SE-7, No. 4, July 1981, pp. 417-426.
- [5] Misra, J., Chandy, Mani and Smith, Todd, "Proving Safety and Liveness of Communicating Processes with Examples," Proceedings of the ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, August 1982, Toronto, Canada.
- [6] Misra, J., "Detecting Termination of Distributed Computations Using Markers," Proceedings of the Second ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, August 17-19, 1983, Montreal, Canada.
- [7] Chandy, Mani and Lamport, Leslie, "Detecting Stability in Distributed Systems," in preparation.
- [8] Chandy, Mani and Misra, J., "Preserving Asymmetry by Symmetric Processes and Distributed Fair Conflict Resolution," submitted to ACM Transactions on Programming Languages and Systems.
- [9] Ossefort, Marty, "A Unified Approach to Formal Verification of Network Safety Properties," Ph.D Thesis, Computer Sciences Dept., The University of Texas, Austin, Texas 78712 (August 1982).
- [10] Ossefort, Marty, "A Formal Proof of a Well-Known Distributed Data Base Update Protocol," to appear in ACM Transactions on Programming Languages and Systems.
- [11] Kumar, Devendra, "Efficient Distributed Simulation Schemes," Ph.D Thesis (in preparation).
- [12] Chandy, Mani and Herman, Ted, "A Distributed Procedure to Detect AND/OR Deadlock," submitted to ACM Transactions on Distributed Systems.

- [13] Misra, J. and Chandy, Mani, "Asynchronous Distributed Simulation Via Deadlock and Recovery," Communications of the ACM, Vol. 24, No. 4, April 1981, pp. 198-205.
- [14] Chandy, Mani and Misra, J., "Distributed Simulation: A Study in Concurrent Program Specification, Design and Proof," IEEE Transactions on Software Engineering, Vol. SE-5, No. 5, Sept. 1979, pp. 440-452.
- [15] Lamport, Leslie, "Time, Clocks, and the Ordering of Events in a Distributed System," Communications of the ACM, Vol. 21, No. 7, July 1978, pp. 558-565.
- [16] Jefferson, D. R. and Sowizral, H. A., "Fast Concurrent Simulation Using the Time Warp Mechanism, Part I: Local Control," Technical Report, The Rand Corporation, Santa Monica, California, July 1982.
- [17] Hudak, Paul, "Distributed Task and Memory Management," Proceedings of the Second ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, August 17-19, 1983, Montreal, Canada.
- [18] Ricart, Glenn and Agrawala, Ashok, "An Optimal Algorithm for Mutual Exclusion in Computer Networks," Communications of the ACM, Vol. 24, No. 1, January 1981, pp. 9-17.
- [19] Dijkstra, E. W. D. "Two Starvation Free Solutions of a General Exclusion Problem," EWD 625, Planaanstraat 5, 5671 AL Nuenen, The Netherlands.
- [20] Obermarck, R. "Distributed Deadlock Detection Algorithms," ACM TODS, Vol. 7, No. 2, June 1982.
- [21] Herman, Ted, Ph.D. Thesis (in preparation), Computer Sciences Department, University of Texas, Austin, Texas 78712.
- [22] LeLann, Gerard, "Distributed Systems - Towards a Formal Approach," Information Processing 77, B. Gilchrist, Editor, IFIP, North-Holland Publishing Company (1977).
- [23] Lehmann, D and Rabin, M., "On the Advantages of Free Choice: A Symmetric and Fully Distributed Solution to the Dining Philosophers Problem," Proceedings of the Eighth Annual ACM Symposium on Principles of Programming Languages, Williamsburg, Virginia, January 26-28, 1981.
- [24] Ossefort, Marty, "Proving Safety Properties for a General Communication Protocol," Proceedings of the ACM SIGCOMM '83 Symposium on Communications, Architectures and Protocols, Austin, Texas, March 7-9, 1983.

- [25] Barringer, Howard and Jones, Clifford B., "A Survey of Verification Techniques for Parallel Programs," Technical Report, Department of Computer Science, University of Manchester, Manchester, England, September 1982.
- [26] Menasce, Daniel and Muntz, Richard, "Locking and Deadlock Detection in Distributed Data Bases," IEEE Transactions on Software Engineering, Vol. SE-5, No. 3, May 1979, pp. 195-202.
- [27] Gligor, Virgil and Shattuck, Susan, "On Deadlock Detection in Distributed Systems," IEEE Transactions on Software Engineering, Vol. SE-6, No. 5, September 1980, pp. 435-439.
- [28] Dijkstra, E. W. and Scholten, C.S., "Termination Detection for Diffusing Computation," Information Processing Letters, Vol. 11, No. 1, August 1980, pp. 1-4.
- [29] Williams, Elizabeth, "Design, Analysis and Implementation of Distributed Systems from a Performance Perspective," Ph.D Thesis, Computer Sciences Department, University of Texas at Austin, Texas 78712, May 1983.

Update to Last Year's Annual Report (see attached)

Item 1. Distributed Computation on Graphs: Shortest Path Algorithms,

Appeared: Communications of the ACM
Volume 25
Number 11
November 1982
pp. 833-837

Item 2. A Distributed Deadlock Detection Algorithms and Its Correctness Proof,

Appeared: New title: Distributed Deadlock Detection
ACM Transactions on Computer Systems
Volume 1
Number 2
May 1983
pp. 144-156

Item 3. A Distributed Graph Algorithm: Knot Detection

Appeared: ACM Transactions on Programming Languages and Systems
Volume 4
Number 4
October 1982
pp. 678-686

List of Publications (1981-82)

1. Distributed Computation on Graphs: Shortest Path Algorithms, to appear in Communications of the ACM, (K. M. Chandy and J. Misra)
2. A Distributed Deadlock Detection Algorithm and Its Correctness Proof, to appear in Communications of the ACM, (K. M. Chandy, J. Misra and L. Haas)
3. A Distributed Graph Algorithm: Knot Detection, to appear in ACM Transactions on Programming Languages and Systems, (J. Misra and K. M. Chandy)
4. A Distributed Algorithm for Detecting Resource Deadlocks in Distributed Systems, Proceedings of the ACM SIGACT-SIGOPS Conference on the Principles of Distributed Computing, August 18-20, 1982, Ottawa, Canada. (K. M. Chandy and J. Misra)
5. Proving Safety and Liveness of Communicating Processes with Examples, Proceedings of the ACM SIGACT-SIGOPS Conference on the Principles of Distributed Computing, August 18-20, 1982, Ottawa, Canada (J. Misra, K. M. Chandy and Todd Smith)

List of Publications (since Annual Report of August 1982)

1. "Finding Repeated Elements," Science of Computer Programming, No. 2, (1982), pp. 143-152, North-Holland Publishing Company. (Jayadev Misra and David Gries)
2. "Assigning Processes to Processors in Distributed Systems," Proceedings of the 1983 International Conference on Parallel Processing, Bellaire, Michigan, August 23-26, 1983, (Elizabeth Williams)
3. "Preserving Asymmetry by Symmetric Processes and Distributed Fair Conflict Resolution," submitted to ACM Transactions on Programming Languages and Systems, (K. Mani Chandy and Jayadev Misra).
4. "A Distributed Procedure to Detect AND/OR Deadlock," submitted to ACM Transactions on Distributed Systems, (K. Mani Chandy and Ted Herman).
5. "Detecting Stability in Distributed Systems," in preparation, (K. Mani Chandy and Leslie Lamport).
6. Design, Analysis and Implementation of Distributed Systems From a Performance Perspective, Ph.D Thesis, Department of Computer Sciences, University of Texas, Austin, Texas 78712, (Elizabeth Williams)
7. Efficient Distributed Simulation Schemes, Ph.D Thesis, (in preparation), Computer Sciences Department, University of Texas, Austin, Texas 78712, (Devendra Kumar).

September, 1983

List of Professional Personnel

Professor K. Mani Chandy, Co-Principal Investigator

Professor Jayadev Misra, Co-Principal Investigator

with

Faculty
Computer Sciences Department
University of Texas
Austin, Texas 78712

List of Ph.D Students (completing degrees)

Elizabeth Williams, degree completed: May, 1983, Ph.D Thesis title:
"Design, Analysis, and Implementation of Distributed Systems from
a Performance Perspective," Computer Sciences Department, University
of Texas, Austin 78712.